Musical Genre Classification
Machine Learning CS 5350/6350, Fall 2016

Sonam Choudhary and Aarushi Sarbhai

Abstract— **In this project we did an experimental research to investigate how our proposed method performs when trying to classify songs by genre. We also study what features are effective in classifying the songs by taking them individually and then in combinations. Each genre is also separately taken to see how the classifier performs in identifying each genre effectively. By using machine learning techniques such as k-NN and k-means, we were able to get some interesting relation between different features and different genres.**

I. INTRODUCTION

Genre classification is a central topic in information retrieval from music and has various upcoming applications. Most significant of those is the use of genre classification in song recommendations. Most web applications used today, such as Spotify and Pandora, manually tag genres to the songs in their database [1]. Whereas others such as YouTube, which have a wider variety of data, lack the metadata required to search for them efficiently. Music Information Retrieval (MIR) has become common and popular due to easy instant access to songs that were previously only available on physical media. Thus one of the main tasks in MIR is to design approaches to find algorithmic implementations to manage such large collections of songs. A common approach is to identify and extract relevant metadata to be used in automatic tag annotation, song recommendation, playlist generation, etc. This is still a challenging task as even humans cannot classify music in genres with absolute or even close to perfect certainty.

In our project one of the features we use to make predictions is Mel-frequency cepstral coefficients (MFCCs). Using MFCC as a feature proved to be very tricky as it involves dealing with high dimensional data. To deal with this feature, we reduced the dimensions by following the method used by John Cast et al. [2] while retaining meaningful components from it. We used a set of 59,600 songs from the Million Songs Dataset. Due to the high dimensionality of data the main aim was to reduce the dimensionality while retaining the meaningful components of the data.

The technique uses audio features, such as loudness, tempo, etc., extracted as discrete values to train sub-classifiers. These are taken individually as week classifiers and tested on parts of training data, using cross validation. The sub-classifiers themselves are k-nearest neighbours (k-NN) and k-means. Cross validation is done on these individual sub-classifiers too. Finally according to the error from cross validation on these sub-classifiers, each is assigned a vote. This vote is nothing but percentage of the contribution of the sub-classifier to the final prediction.

II. RELATED WORK

A lot of work has been done in the past in music genre classification using different techniques. Although music genre tagging by humans is some what successful but it is hoped that machines can be incorporated to do this job with much less effort and more accurately by analysing certain attributes of each song. Earlier work included genre classification using Signal processing processing techniques to find low-level features that correlate with musical genre. Using Gaussian mixture models and diagonal covariance matrices, George Tzanetakis and Perry Cook achieved 61 percent classification accuracy with ten genres [3]. The 3 features they used for classification were timbrel texture, rhythmic content, and pitch content. Their results were comparable to those of human classification, although not quite as good.

Most methods for machine tagging genres combine features from different sources into a larger dataset to represent a song [4]. Due to varied sources, there has been mismatch in features. Using a huge database with a very large amount of features may lead to overfitting. It is also not possible to get valid data from each source for each instance.

A number of online music services exist such as Pandora, Last.fm, and Spotify, which are successful but most of these are merely based on traditional text information retrieval.

III. Methodology

The technique used to predict labels uses a set of weak classifiers or sub-classifiers by taking each feature individually. This is done because there is no study of how much a certain feature effects the prediction of the genres. The sub-classifiers are k-NN and k-mean classifiers.

First, we analyze how each sub-classifier performs on a part of the training data to try to estimate how useful the classifier is in predicting the genres. According to the error in this classification, each sub-classifier is assigned weights. The final classification is done using this multi-class Adaboost classifier [5].

The accuracy is significantly improved by restricting the predictions to the genres that the artist is listed under from the dataset. We do not include this in the final prediction as there are very few artists that come under multiple genres.

IV. Implementation

A. Data Collection

We had originally decided to use sheet music to do genre classification. However, the kind of information that can be extracted from sheet music (image) was not sufficient to get a holistic view of different dimensions required to classify music. It would, in most cases, be possible to know the different instruments involved, the key and mode for the song. Along with this information the timing of these notes plays a very important role. Not only could this information not be extracted from sheet music, it also makes the data time variant making the dataset multidimensional over time. A trained musician would need features like beat, chord progressions and distinct instruments to classify songs. Although he would be able to extract such features from audio files, an algorithmic classification cannot use the same identifiers. It is not possible to use signal processing techniques to reliably detect these features in such audio files. Due to difficulties in relevant feature extraction from the mentioned sources, we took the dataset from the Million Songs Dataset [6]. The entire dataset contains 1,000,000 tracks from 44,745 unique artists. To begin with we are using a subset of songs and splitting them by artist. To begin with we are using a subset of 59,600 songs. These are also split by genre to analyze how each of the features we are taking will contribute to the genre prediction. We are taking 70% of the songs in the training set and the rest 30% in the test set. The dataset contains basic metadata and audio features for songs, each identified by a track ID. Some of the features are self explanatory such as the artist name, song title, loudness, tempo, duration (in seconds), mode or scale, and key. Some other features are - time signature which represents the length of the melody and timbre which gives the tone quality. Time signatures are written as fractions with the numerator denoting the number of beats in a measure and the denominator denotes the note value which makes one beat (usually 4). The timbre is represented in terms of 12 Mel-Frequency Cepstrum (MFC) coefficients. We use basic domain knowledge and use features such as loudness, tempo, key, mode, and timbre values as MFC coefficients. There are 10 genres tyoes that our classifier tries to predict, namely, classic pop and rock, punk, folk, pop, dance and electronica, metal, jazz and blues, classical, hip-hop, soul and reggae.

B. Timbral Analysis

MFCC is one of the key features that play an important role in the classification. They are scaled to a humans perception of sound [3]. The 12 coefficients represent different metadata about the sound. As shown in the work of T. Camenzind and S. Goel [7] we model the data values using a Multivariate Gaussian Distribution. We calculate the Maximum Likelihood Estimation Gaussian for each song, and represent the song using the mean μ and covariance σ of this distribution. Figure 1 shows how this data is extracted from an audio signal. Each segment is 250ms long and on an average a song has 1000 such segments. We get 12 coefficients for each segment which are reduced by modelling them into the Gaussian Distribution.

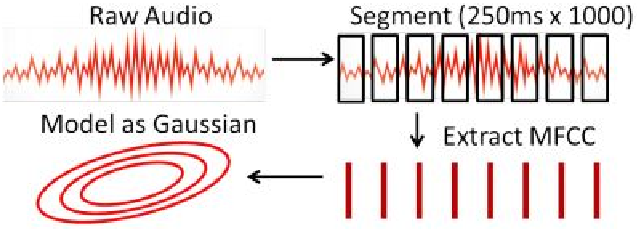

Figure 1 : MFCC Extraction and the Gaussian Model

C. Experiment

The k-NN and k-means sub-classifiers take 90% of the training data for training in the cross-validation phase. The rest of the training data is used as the testing data in the training phase to get an idea of how much error to expect. Although the main motivation of doing cross-validation was to converge on a value of k, we did not see any significant difference for different values of k. The accuracies observed by the individual features using sub-classifiers on the test data is given in Figure 2.

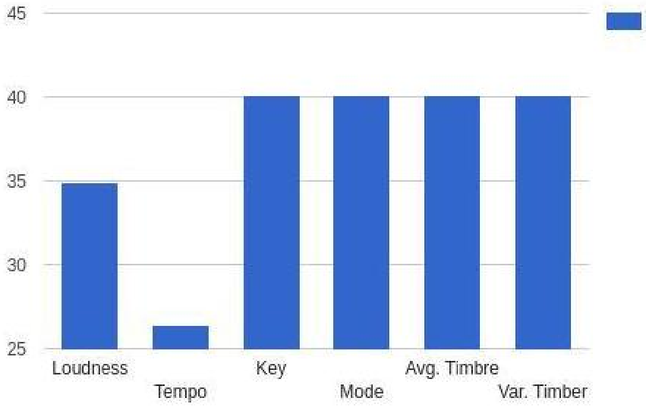

Figure 2 (a) : Accuracy from k-NN on test data

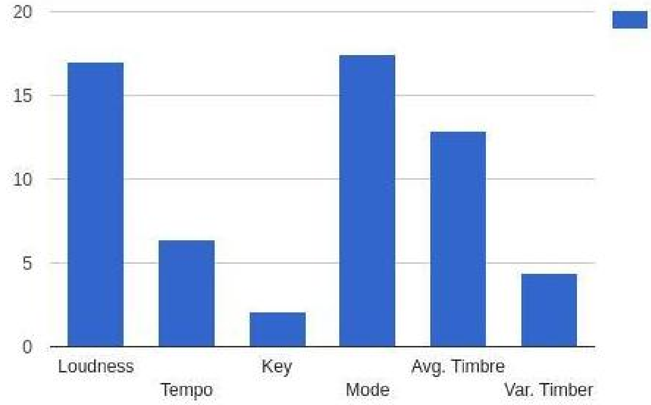

Figure 2 (b) : Accuracy from k-means on test data

All these sub-classifiers are taken as weak classifiers for the Adaboost algorithm. The final prediction is made by taking the weighted predictions of these weak classifiers.

We also try to analyze how well each genre can be predicted given the features we've selected. The results obtained for each genre are shown in Figure 3.

V. RESULTS

The baseline accuracy for our data is 40%. This is the case when all the labels are classified as highest occurring genre, classic pop and rock.As shown in Figure 2, the results from k-NN was better than k-means. Thus the weights of the k-NN classifiers was higher than k-means classifiers. Without shuffling results the prediction made by the classifier is just above the baseline at 42.44%. With shuffling data we get better accuracy of 58.88%. The best result is obtained when artist information is included to restrict genre space. The accuracy in this case rises to 96.47%. We disregard this result as this is because our dataset doesn't include an exhaustive selection of genres performed by the artist.

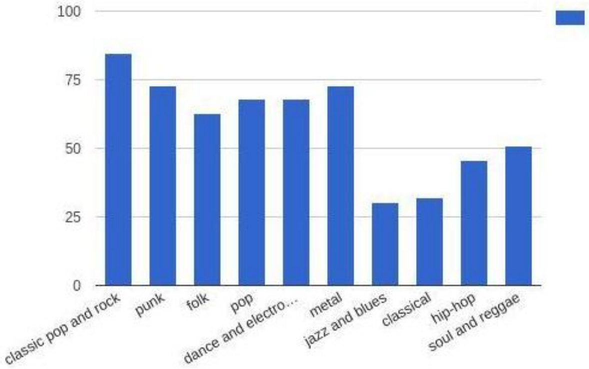

Figure 3 : Accuracy of individual genre

VI. CONCLUSION AND FUTURE WORK

We tried to predict the genre of songs using a new technique. We also tried to study the influence of individual features that we used to classify the songs, which validated our domain knowledge. We also tried to analyze how each of the classifier performs for each of the genres. To improve the results of our technique, we can use the entire dataset of a million songs. If we exclude some of the features which do not give good accuracies then we can get better results by avoiding overfitting. We can also include other useful features not included in this database. The MFCC values are all aggregated, instead of taking the 12 values separately. Intuitively, taking the values separately should improve results as they all denote different kinds of metadata and are in different ranges.

REFERENCES

[1] Y. Panagakis and C. Kotropoulos, "Music classification by low-rank semantic mappings," *EURASIP Journal on Audio, Speech and Music Processing*, 2013.

[2] J. Cast, C. Schulze, and A. Fauci, "Music genre classification," aug 2013.

[3] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10(5), jul 2002.

[4] C. McKay, J. A. Burgoyne, J. Hockman, J. B. L. Smith, G. Vigliensoni, and I. Fujinaga, "Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features," *Proceedings of the 11th ISMIR, ISMIR*, p. 213âĂŞ718, 2010.

[5] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class adaboost," *Statistics and its Interface*, aug 2009.

[6] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "Million song dataset - http://labrosa.ee.columbia.edu/millionsong/," *ISMIRC*, 2011.

[7] T. Camenzind and S. Goel, "jazz : Automatic music genre detection," *Stanford University, Machine Learning Final Projects*, aug 2013.